



Licenciatura Engenharia Informática e Multimédia

Programação de Conteúdos Multimédia - PCM

Relatório Projeto Final

Docente: Eng. Rui Jesus

Trabalho realizado por:

Fábio Dias, nº 42921

1. Introdução

Para este projecto final, foi-nos pedido a realização de um clone do Google Images, com base nos ficheiros disponibilizados pelo docente.

Foi realizada uma versão com a procura de imagens a partir de palavras-chave, assim como, por semelhança de cores, utilizando a cor dominante de cada imagem.

2. Desenvolvimento

2.1. HTML

As partes fundamentais de este projecto são a barra de pesquisa, onde se colocam as palavra-chave a serem pesquisadas; a barra de cores, onde estão as cores dominantes que o browser mostra; e o *canvas*, onde são apresentadas as imagens, tanto por palavra-chave como por cores.

Utilizando o ficheiro *html* fornecido pelo docente, no *head* importei o Bootstrap 5, tal como se pode observar na Fig. 1. Para o *body*, criei um *header* (Fig. 2), onde inseri o logotipo deste clone do Google Images, uma barra de pesquisa com as cores todas alinhadas e um botão para a pesquisa.

Mais abaixo, mas ainda dentro do *body*, encontra-se o *canvas*, onde as imagens são apresentadas. Por fim, a função *main*, explicada mais à frente, é chamada quando o browser acaba de carregar esta página. Tudo isto pode ser observado na Fig. 3.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="PCM Final Project - Visual Search">
  <meta name="keywords" content="HTML5, CSS, JavaScript">
  <meta name="author" content="Rui Jesus">
  <meta charset="UTF-8">
  <title>Visual Search</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3Q0" >
  <link rel="stylesheet" href="css/VSearchStyle.css">

  <script src="js/Image_Processing.js"></script>
  <script src="js/XML_Database.js"></script>
  <script src="js/ISearchEngine.js"></script>
  <script src="js/ISearchEngine_Manager.js"></script>
</head>
```

Figura 1 - Head no HTML

```
<body>
<header class="container" id="header1">
  <div class="row pt-2">
    
    <div class="col-8">
      <div class="row">
        <input id="searchBar" type="search">
      </div>
      <div id="colorPicker" class="row mt-2" style="height: 100%; margin-bottom: 0px !important;">
        <div id="red" class="col-1 rounded-circle" style="background-color: rgb(204, 0, 0);"></div>
        <div id="orange" class="col-1 rounded-circle" style="background-color: rgb(251, 148, 11);"></div>
        <div id="yellow" class="col-1 rounded-circle" style="background-color: rgb(255, 255, 0);"></div>
        <div id="green" class="col-1 rounded-circle" style="background-color: rgb(0, 204, 0);"></div>
        <div id="Blue-green" class="col-1 rounded-circle" style="background-color: rgb(3, 192, 198);"></div>
        <div id="blue" class="col-1 rounded-circle" style="background-color: rgb(0, 0, 255);"></div>
        <div id="purple" class="col-1 rounded-circle" style="background-color: rgb(118, 44, 167);"></div>
        <div id="pink" class="col-1 rounded-circle" style="background-color: rgb(255, 152, 191);"></div>
        <div id="white" class="col-1 rounded-circle" style="background-color: rgb(255, 255, 255);"></div>
        <div id="grey" class="col-1 rounded-circle" style="background-color: rgb(153, 153, 153);"></div>
        <div id="black" class="col-1 rounded-circle" style="background-color: rgb(0, 0, 0);"></div>
        <div id="brown" class="col-1 rounded-circle" style="background-color: rgb(136, 84, 24);"></div>
      </div>
    </div>
    <input class="col-1 mb-2 ml-2" id="searchButton" type="button">
    <input class="col-1 mb-2 ml-2" id="toggleDarkLightMode" type="button">
  </div>
</header>
```

Figura 2 - Body, Header no HTML

```

<section id="visualizacao" class="mt-5 row justify-content-center">
  <canvas id="canvas" width="1100" height="1100">
    |   Your browser doesn't recognize the canvas element
  </canvas>
</section>

<footer>
  |   <!--<p>Rui Jesus</p>-->
</footer>

<script> window.onload = main(); </script>
</body>
</html>

```

Figura 3 - Canvas no HTML

2.2. Javascript

O fluxo do programa começa no *script ISearchEngine_Manager*. Aqui, no seu método *main* (Fig. 4), é instanciada e iniciada a classe *ISearchEngine* que possui a lista de imagens, assim como as categorias, cores e cada valor dos componentes RGB.

O método *init* possui o *databaseProcessing* (Fig. 5), e deve ser chamado apenas uma vez e, de seguida, comentado.

Este método processa cada imagem e chama o método que permite a criação dos ficheiros XML e a colocação do mesmo no *Local Storage*. Dado que só é necessário correr uma vez, deve-se comentar para poupar processamento, e para permitir o funcionamento normal da aplicação.

Ainda no *main* (Fig. 4), são adicionados vários *Event Listeners*, um para o clique no botão de pesquisa por palavra-chave, e um por cada uma das cores exibidas. Estes chamam os métodos *searchKeywords* e *searchColors*, respectivamente.

O método *searchKeywords* (Fig. 6) pesquisa, no XML local, as imagens que pertencem à categoria passada como argumento, o array das imagens amostradas é limpo e são inseridas as novas imagens encontradas, depois é chamado o método *gridView* que apresenta as imagens com os devidos espaçamentos e alinhamentos. *searchColors* (Fig. 7) pesquisa no XML do *Local Storage* as primeiras imagens que tenham a cor dominante semelhante à cor seleccionada, depois, tal como no *searchKeywords*, é limpo o array, são inseridas estas novas imagens e é chamado o *gridView*.

```

function main() {
    let canvas = document.querySelector("canvas");
    app = new ISearchEngine("XML/Image_database.xml");
    app.init(canvas);

    let searchButton = document.getElementById("searchButton");
    let keywordBar = document.getElementById("searchBar");

    app.colors.forEach(function(colorID) {
        let color = document.getElementById(colorID);

        color.addEventListener('click', function()
        {
            app.searchColor(keywordBar.value.toLowerCase(), colorID, canvas);
        }, false);
    });

    searchButton.addEventListener('click', function()
    {
        app.searchKeywords(keywordBar.value.toLowerCase(), canvas);
    }, false);
}

```

Figura 4 - Método main, no ISearchEngine_Manager

```

databaseProcessing (cnv) {
    let h12color = new ColorHistogram(this.redColor, this.greenColor, this.blueColor);
    let colmoments = new ColorMoments();

    let self = this;

    this.categories.forEach(function(category)
    {
        let paths = self.XML_db.SearchXMLByKeyword(category, self.num_Images);
        let dominantColors = self.XML_db.SearchXMLByColor(category, self.num_Images);

        let index = 0;

        paths.forEach(function(path)
        {
            let img = new Picture(0, 0, 100, 100, path, category);
            img.setDominantColor(dominantColors[index]);
            index++;

            let eventname = "processed_picture_" + img.impath;
            let eventP = new Event(eventname);

            document.addEventListener(eventname, function(){
                self.imageProcessed(img, eventname);
            }, false);

            img.computation(cnv, h12color, colmoments, eventP);
        });
    });
}

```

Figura 5 - Método databaseProcessing, no ISearch_Engine

```

searchKeywords(category, canvas) {
    let newXML = this.XML_db.loadXMLfile(this.XML_file);
    this.XML_db.setXML(newXML);
    //console.log(this.XML_file);

    console.log("1º");
    console.log(this.XML_db.xml);
    let paths = this.XML_db.SearchXMLByKeyword(category, this.numshownpic);

    this.allpictures.empty_Pool();

    for(let i = 0; i < paths.length; i++)
    {
        this.allpictures.insert(new Picture(0, 0, this.imgWidth, this.imgHeight, paths[i], category));
    }

    this.gridView(canvas);
}

```

Figura 6 - Método searchKeywords, no ISearch_Engine

```

searchColor(category, color, canvas) {
  let newXML = this.LS_db.readLS_XML(category);
  this.XML_db.setXML(newXML);
  let paths = this.XML_db.SearchXMLByKeyword(color, this.numshownpic);
  //console.log(this.XML_db.xml);
  this.allpictures.empty_Pool();
  //console.log(paths.length); //LS ou SearchByKeyword
  for(let i = 0; i < paths.length; i++)
  {
    this.allpictures.insert(new Picture(0, 0, this.imgWidth, this.imgHeight, paths[i], category));
  }

  //console.log(this.allpictures.stuff);

  this.gridView(canvas);
}

```

Figura 7 - Método searchColor, no ISearch_Engine

Com esta aplicação implementada, é possível observar o seguinte resultado.

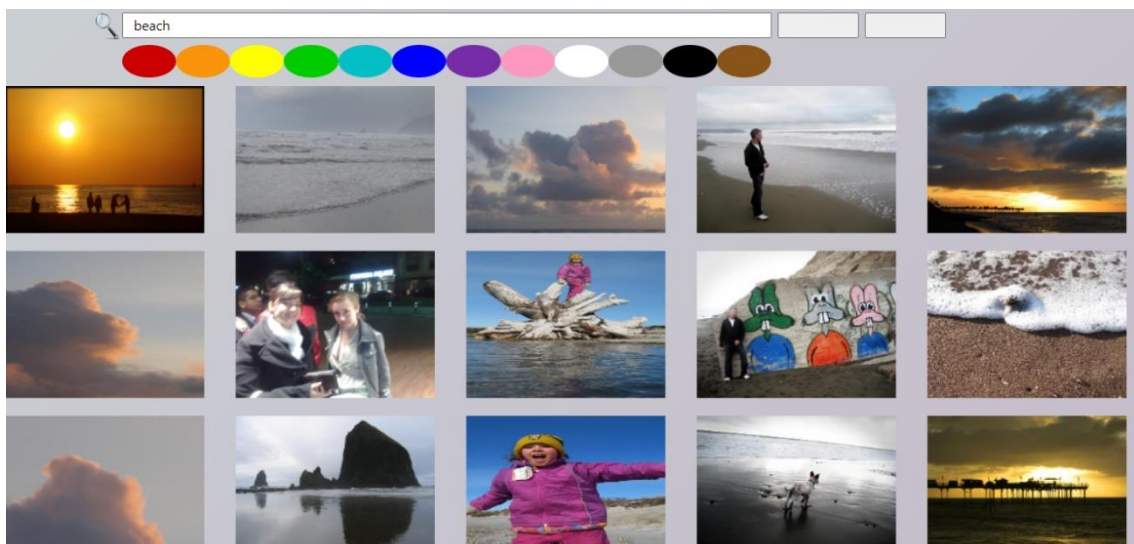


Figura 8 - Interface Gráfica da aplicação

3. Conclusão

Embora este clone do Google Images seja um projecto interessante, tive de apontar para a versão mais simples devido à falta de tempo útil para o desenvolvimento deste programa. Tinha algumas ideias de como tornar a sua interface gráfica mais apelativa, como o canvas estar centrado e possuir um botão que variava entre o *light theme* e o *dark theme*, tema claro e escuro, mas acabou por cair no desuso.

Mesmo assim, foi possível observar uma aplicação que é capaz de pesquisar imagens por base em ficheiros XML, tanto locais, como no *Local Storage*. Para além disso, foi possível filtrar, por entre milhares de imagens, por *queries* diferentes, como palavras-chave e cor,